

IN THE CLAIMS

1. (Currently Amended) A method of dynamically controlling restart processing for recovery of an application, said method comprising the steps of:

dynamically calculating a restart time required to restart a resource manager in case of termination or recovery of said resource manager; and

dynamically initiating said resource manager to take a checkpoint whenever said restart time exceeds a specified [illegible].

2. (Currently Amended) The [illegible]ethod of claim 1 further comprising the step of receiving a new request to modify resource.

3. (Currently Amended) The [illegible]ethod of claim 2, wherein the step of dynamically calculating comprises calculating said restart time to include the new request and all accumulated requests since a last checkpoint which participate in the restart processing.

4. (Currently Amended) The [illegible]ethod of claim 2, wherein the specified restart time is specified to said resource manager by said application.

5. (Currently Amended) The [illegible]ethod of claim 4, wherein said application comprises an application client.

6. (Currently Amended) The [illegible]ethod of claim 4, wherein said application comprises an application server.

7. (Currently Amended) The method of claim 6, wherein said application server also administrates a request-load-log of all accumulated requests since said last checkpoint for the step of dynamically calculating said restart time.

8. (Currently Amended) The method of claim 6, wherein a plurality of resource managers are connected to said application server.

9. (Currently Amended) The method of claim 6, wherein an application client specifies to said application server a maximum request response time defining a time not to be exceeded for application requests even in case of termination or recovery of said resource manager, and wherein said application server converts said maximum request response time to said current maximum restart time by subtracting from said maximum request response time processing time of further activities associated with said restart processing.

10. (Currently Amended) The method of claim 9, wherein said further activities comprise processing time required to restart said application server.

11. (Currently Amended) The method of claim 9, wherein said further activities comprise processing time required for re-attaching said application server to said resource manager.

12. (Currently Amended) The method of claim 1, wherein a resource manager performs the steps of dynamically calculating and dynamically initiating.

13. (Currently Amended) The method of claim 1, wherein an application server performs the steps of dynamically calculating and dynamically initiating.

14. (Original) A method of recovering a valid state of a resource manager in response to an abnormal termination of the resource manager, the method comprising the steps of:

receiving a plurality of requests for one or more resources managed by the resource manager;

calculating, as each one of the requests is processed, the time required to restore a valid state of the resource manager in response to an abnormal termination of the resource manager; and

saving the current state of the resource manager each time the calculated restore time exceeds a specified time period.

15. (Cancelled)

16. (Original) A computer program product stored on a computer usable medium comprising computer readable program code portions for causing a computer to perform a method for dynamically controlling restart processing for recovery of an application, when said computer readable program code portions are executed on a computer, said method comprising the steps of:

dynamically calculating a restart time required to restart a resource manager in case of termination or recovery of said resource manager; and

dynamically initiating said resource manager to take a checkpoint whenever said restart time exceeds a specified restart time.

Please add the following new claims:

17. (New) A method of dynamically controlling restart processing for recovery of an application such as an application server, said method comprising the steps of:

receiving from an application client a maximum request response time defining a time not to be exceeded for application requests even in case of termination or recover of a resource manager;

dynamically calculating a restart time required to restart a resource manager in case of termination or recovery of said resource manager including the time required for processing further activities associated with the restart process; and

dynamically initiating said resource manager to take a checkpoint whenever said restart time exceeds the maximum request response time.

18. (New) The method of claim 17, wherein said further activities comprise processing time required to restart said application server.

19. (New) The method of claim 17, wherein said further activities comprise processing time required for re-attaching said application server to said resource manager.

20. (New) A method of dynamically controlling restart processing for recovery of an application, said method comprising the steps of:

dynamically calculating a restart time required to restart a resource manager in case of termination or recovery of said resource manager; and

dynamically initiating said resource manager to take a checkpoint whenever said restart time exceeds a period of time specified by a client application.

21. (New) The method of claim 20, wherein the step of dynamically calculating comprises calculating said restart time to include the new request and all accumulated requests since a last checkpoint which participate in the restart processing.

22. (New) A computer program product comprising a computer useable medium including a computer readable program for dynamically controlling the restart processing for recovery of an application such as an application server, wherein the computer readable program when executed on a computer causes the computer to:

receive from an application client a maximum request response time defining a time not to be exceeded for application requests even in case of termination or recover of a resource manager;

dynamically calculate a restart time required to restart a resource manager in case of termination or recovery of said resource manager including the time required for processing further activities associated with the restart process; and

dynamically initiating said resource manager to take a checkpoint whenever said restart time exceeds the maximum request response time;

23. (New) The computer program product of claim 22, wherein further activities includes the processing time required to restart said application server.

24. (New) The computer program product of claim 22, wherein said further activities comprise processing time required for re-attaching said application server to said resource manager.

DRAWINGS

On Page 2, paragraph 2, of the current Office Action, the Examiner objected to Figure 5, because designation "530" was not referenced in the specification. The specification has been amended to include this reference as noted in this response.

AMENDMENT TO THE SPECIFICATION

Please amend the specification, as originally filed, on page 11 paragraph [0051] as noted below:

"FIG. 5 shows the actions that the application server needs to take when the resource manager (RM) supports a user-defined restart time. In step 500, the application server obtains user specified maximum request response time, hereafter referred to as request response time. The application server then performs a set of actions (indicated by loop_[illegible]) for all resource managers. Step 510 calculates the appropriate restart time for the resource manager from the specified request response time. Transformation of the request response time to the restart time of the resource manager is necessary to cope with additional processing needs, such as the start up of the application server itself. In step 520, the application server hands over this restart time to the resource manager. The resource manager itself would then execute the code shown in FIG. 3 if called by the application server. It should be noted, that FIG. 5 is for illustration purpose only; actual implementations are most likely more sophisticated."

REJECTION OF CLAIMS

Claim 15 was rejected under 35 U.S.C. Section 101. Claim 15 has been cancelled.

Claims 1-8 and 12-16 were rejected under 35 U.S.C. Section 102(e) as being anticipated by U.S. Patent No. 6,351,754 to Bridge, Jr, et. Al ("Bridge").

Bridge

Bridge discloses a method and system for reducing the overhead associated with the recovery from a failure. More specifically, Bridge discloses the concept of using a predetermined count of records (redo records) to determine when a checkpoint should be initiated (See Col 5. Lines 55-60).